Domain-Independent Heuristics for Goal Formulation

Mark Wilson[1], Matthew Molineaux[2], and David W. Aha[1]

[1]Navy Center for Applied Research in Artificial Intelligence; Naval Research Laboratory; Washington, DC

[2]Knexus Research Corporation; Springfield, VA

first.last@nrl.navy.mil | matthew.molineaux@knexusresearch.com

Abstract

Goal-driven autonomy is a framework for intelligent agents that automatically formulate and manage goals in dynamic environments, where goal formulation is the task of identifying goals that the agent should attempt to achieve. We argue that goal formulation is central to high-level autonomy, and explain why identifying domain-independent heuristics for this task is an important research topic in high-level control. We describe two novel domain-independent heuristics for goal formulation (*motivators*) that evaluate the utility of goals based on the projected consequences of achieving them. We then describe their integration in M-ARTUE, an agent that balances the satisfaction of internal needs with the achievement of goals introduced externally. We assess its performance in a series of experiments in the Rovers With Compass domain. Our results show that using domain-independent heuristics yields performance comparable to using domain-specific knowledge for goal formulation. Finally, in ablation studies we demonstrate that each motivator contributes significantly to M-ARTUE's performance.

1. Introduction

An interesting property of human intelligence is that people do not always do what they're told. Those who inflexibly follow given instructions may be accused of lacking initiative, or they may frustrate others by failing to make changes where obviously necessary. Most autonomous agents are frustrating in the same way; even those that can dynamically replan attempt to achieve only those goals that they were given. Avoiding this type of behavior requires an agent to form its own goals (i.e., perform *goal formulation*).

Goal formulation is pivotal to research on high-level agent control because it addresses an important problem: most agents fail to act reasonably when they encounter new, unexpected situations. Agents without goal formulation are limited to doing what they are told because

their motivation is supplied only by an external source. Conversely, agents with an internal source of motivation can go beyond their instructions. For example, if a UAV capable of autonomous goal formulation burns fuel at unexpectedly high rates (because it is flying into the wind), it would formulate a goal to refuel. In contrast, a replanning system might attempt to replan for its original goal without understanding the increased consumption, which may be disastrous. As another example, suppose a transport agent spots a new, unexplored path while on a delivery. A goal-formulating agent might pursue knowledge by following the unexplored path, even though this investigation could delay the completion of its current task. Goal formulation allows the agent to consider *internal* needs rather than only those goals assigned by a human.

Goal formulation is more robust than replanning because it can automatically generate good behaviors in situations the designer did not consider. An agent designer who wished to encode a desire to explore could, for example, add constraints or additional goals to the current problem. However, this requires specifying constraints or goals (for linear planners), or many additional decompositions (for hierarchical planners), for all possible scenarios. We instead study agents that take the initiative in the absence of specific, exhaustive instructions.

Goal formulation is a key process in the goal-driven autonomy (GDA) framework for autonomous agents (Klenk *et al*. 2012). GDA agents perform a cycle of planning and execution monitoring that includes a goal formulation step (see Section 3). Some prior work on GDA has used domain-dependent knowledge for goal formulation. For example, ARTUE (Molineaux *et al*. 2010a) uses manually-engineered trigger rules that add goals suggested by a rule when its trigger conditions are met. This reactive mechanism allows the designer to instruct and control ARTUE, but prevents it from leveraging its own experience, responding to changing environments, or acting robustly in situations where its knowledge is incomplete.

We address this limitation by extending ARTUE with an ability to formulate goals without domain-specific rules and refer to this agent as *Motivated Autonomous Response*

Report Documentation Page

1. REPORT DATE **MAY 2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE **Domain-Independent Heuristics for Goal Formulation**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Navy Center for Applied Research in Artificial Intelligence,Naval Research Laboratory,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, 22-24 May 2013, St, Pete Beach, FL. U.S. Government or Federal Rights License

14. ABSTRACT
Goal-driven autonomy is a framework for intelligent agents that automatically formulate and manage goals in dynamic environments, where goal formulation is the task of identifying goals that the agent should attempt to achieve. We argue that goal formulation is central to high-level autonomy, and explain why identifying domain-independent heuristics for this task is an important research topic in highlevel control. We describe two novel domain-independent heuristics for goal formulation (motivators) that evaluate the utility of goals based on the projected consequences of achieving them. We then describe their integration in MARTUE an agent that balances the satisfaction of internal needs with the achievement of goals introduced externally. We assess its performance in a series of experiments in the Rovers With Compass domain. Our results show that using domain-independent heuristics yields performance comparable to using domain-specific knowledge for goal formulation. Finally, in ablation studies we demonstrate that each motivator contributes significantly to M-ARTUE?s performance.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

to Unexpected Events (M-ARTUE). This initial version of M-ARTUE performs goal formulation using three competing sources of goals, called *motivators*. The *Opportunity* and *Exploration Motivators* choose goals that meet the domain-independent needs of an intelligent agent, while the *Social Motivator* chooses goals that are provided externally (i.e., by a human). Each motivator computes a scaled value denoting the urgency of the need it represents, and chooses a goal that best meets its needs. These motivators are inspired by psychological notions of drives that are independent of particular tasks. We do not claim that they are exhaustive of all possible such motivators, but that they are sufficient to perform comparably to rule-based formulation in some interesting domains.

We claim that a system that formulates goals using domain-independent heuristics can, in some cases, perform comparably to a system guided by expert knowledge, without the cost of engineering that knowledge. To support this claim, we describe a study that compares the performance of ARTUE (using rule-based goal formulation) with M-ARTUE. This study reports results for only a single domain, but the techniques are domain-independent and its lessons can be extended to future studies.

Section 2 summarizes related work in goal formulation and motivation. Section 3 describes the GDA model and its instantiation in ARTUE. Section 4 details the three motivators and a motivation manager that mediates among them. Section 5 describes our investigation in which we analyze M-ARTUE's performance. Finally, Section 6 concludes with a discussion of limitations and future work.

2.   Related Work

While domain-independent goal formulation heuristics have not previously been examined, research on goal formulation and management dates back to Bratman's (1987) introduction of the BDI model. We summarize prior related work on agents that *explicitly* perform goal formulation and management tasks. Most related research does not consider the problem of domain-independent heuristics for goal formulation; two systems (T-ARTUE and EISBot) learn goal formulation knowledge based on human expertise, but do not consider internal needs of the agent and must re-learn for new domains.

Some cognitive architectures perform goal formulation. For example, ICARUS (Choi 2011) uses a *reactive* goal management procedure to nominate and prioritize new top-level goals in which <condition, goal> pairs in long-term goal memory are considered for nomination at every reasoning step. This resembles rule-based goal-formulation approaches, as seen in ARTUE. CLARION (Sun 2007) instead includes a motivation subsystem that formulates goals based on the psychological notion of *drives*, which constitute a hierarchy of heuristic functions. This resembles M-ARTUE's approach because it supports internal and external needs. However, the representation of these needs is domain dependent.

Section 3 details earlier work on ARTUE (Molineaux *et al*. 2010a). M-ARTUE differs only in the way goals are formulated; instead of using reactive rules, it uses domain-independent heuristics to evaluate potential goals.

In realistic domains it is often infeasible to provide goal formulation knowledge for every situation. To address this, T-ARTUE (Powell *et al*. 2011) and EISBot (Weber *et al*. 2012) learn this knowledge from humans: T-ARTUE learns from criticism and answers to queries, while EISBot learns from human demonstrations. Each provides a domain-independent method for acquiring formulation knowledge, but neither system reasons about internal needs alongside external goals.

Although based on the GDA model, LGDA (Jaidee *et al*. 2011) differs substantially from ARTUE and M-ARTUE. LGDA learns its goal selection function using Q-learning. While this increases autonomy, it employs a domain-dependent reward function; indirectly, LGDA's goal selection strategy is guided by a human.

MADBot (Coddington 2006) represents motivations using domain knowledge to encode thresholds for known variables that the agent can observe. In contrast, M-ARTUE does not represent motivations using domain knowledge, and is not limited to generating goals for achieving threshold values.

Dora the Explorer (Hawes *et al*. 2011) encodes motivators that formulate goals related to exploring space and determining the function of rooms, similar to M-ARTUE's exploration motivator. However, Dora's functions are domain-specific. Finally, Hawes's (2011) survey of motivation frameworks defines goal management and goal formulation in terms of *goal generators* or *drives*. It relates many systems in terms of these concepts, and proposes a design for future "motive management frameworks". M-ARTUE satisfies several requirements of this design, including the use of a continual planner, goal generators independent of the planning process (i.e., motivators), and concepts of urgency and fitness that match the requirements of *importance* and *urgency* fairly closely. However, we ignore the issue of oversubscription planning, and where Hawes's design calls for motivations to be represented as resources, we instead treat resources as a *source* of motivation. In contrast to Hawes's framework, M-ARTUE has an additional design constraint that specifies motivators to formulate goals in a domain-*independent* manner, without hand-engineered knowledge. This allows agents that can adapt to changing environments to react robustly in ways the designer did not pre-specify.

3.   GDA and ARTUE

Goal-Driven Autonomy (GDA) is a conceptual model for online planning in autonomous agents (Klenk *et al.* 2012). It separates the planning process from procedures for goal formulation and management. ARTUE implements a GDA model that has been demonstrated previously in complex environments (Molineaux *et al.* 2012).

3.1 GDA Conceptual Model

Figure 1 illustrates how GDA extends Nau's (2007) model of online planning. The GDA model expands and details the *Controller*, which interacts with a *Planner* and a *State Transition System* Σ (an execution environment).

System Σ is a tuple (S,A,F,γ) with states S, actions A, exogenous events F, and state transition function γ: $S \times (A \cup F) \rightarrow S$, which describes how an action's execution (or an event's occurrence) transforms the environment from one state to another. In complex environments, the agent has only partial access to S, F, and γ.

The Planner receives as input a planning problem (M_Σ, s_c, g_c), where M_Σ is a model of Σ, s_c is the current state, and g_c is a goal, from the set of all possible goals G, that can be satisfied by some set of states $S_g \subseteq S$. The Planner outputs (1) a plan p_c, which is a sequence of actions $A_c = [a_{c+1},...,a_{c+n}]$, and (2) a corresponding sequence of expectations $X_c = [x_{c+1},...x_{c+n}]$, where each $x_i \in X_c$ is the expected state that should follow when executing the corresponding a_i in A_c.

The Controller takes as input initial state s_0, initial goal g_0, and M_Σ, and sends them to the Planner to generate plan p_0 and expectations X_0. The Controller forwards p_0's actions to Σ for execution and processes the resulting observations, where Σ also continually receives and processes any actions from other interacting agents and exogenous events.

During plan execution, the Controller performs the following knowledge-intensive GDA tasks:

Discrepancy detection: GDA detects unexpected events by comparing the observations s_c (obtained from executing action a_c in state s_{c-1}) with expectation $x_c \in X$. If one or more discrepancies $d \in D$ (i.e., the set of possible discrepancies) are found, then explanation generation is performed.

Explanation generation: Given a state s_c and a discrepancy $d \in D$, this task hypothesizes one or more explanations of the discrepancy's cause $e \in E$, the set of possible explanations, allowing the cause to be addressed directly.

Goal formulation: Resolving a discrepancy may warrant a change in the current goal(s). If so, this task formulates a goal $g \in G$ in response to d, given also e and s_c.

Goal management: The formulation of a new goal may warrant its immediate focus and/or removal of some existing goals. Given a set of pending goals $G_P \subseteq G$ and new goal $g \in G$, this task may update G_P (e.g., by adding g and/or deleting/modifying other pending goals) and then select the next goal $g' \in G_P$ to be given to the Planner. (It is possible that $g = g'$.)

GDA makes no commitments to specific types of algorithms for the highlighted tasks (e.g., goal management may involve comprehensive goal transformations (Cox and Veloso 1998)), and treats the Planner as a black box.

3.2 ARTUE Agent

ARTUE (Molineaux *et al*. 2010a) performs discrepancy detection using a set-difference operation over expectations
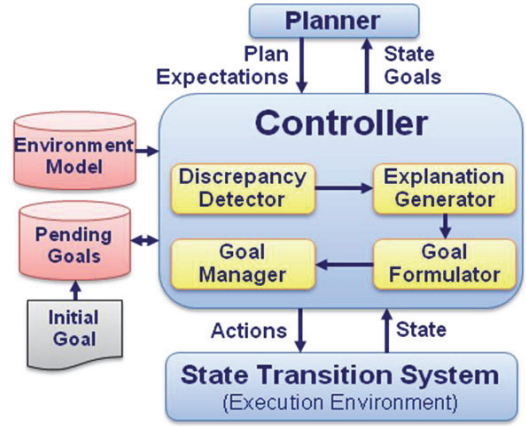

Figure 1: GDA Conceptual Model

and observations, and performs explanation generation using the DiscoverHistory algorithm (Molineaux *et al*. 2012), which creates explanations by hypothesizing events that resolve inconsistencies in the agent's model of occurences. ARTUE uses a version of the hierarchical network (HTN) planner SHOP2 (Nau *et al*. 2003) to generate plans. To predict future events, Molineaux et al. (2010b) extended SHOP2 to reason about planning models that include events in the PDDL+ representation. To work with an HTN planner, we assume that a mapping exists from any goal to an HTN task that can accomplish it. This mapping is performed and the appropriate task, rather than the goal, is given to the planner for plan generation. Goal formulation in ARTUE is based on a set of state-goal trigger rules designed manually by a domain expert. Section 4 describes our new goal formulation techniques for M-ARTUE.

4. Goal Formulation and Management

In this paper, we replace ARTUE's domain-specific rule-based method for goal formulation and management with a heuristic-based formulator that is domain independent. It evaluates potential new goals for M-ARTUE using three *motivators*. Each motivator calculates a value for *urgency* that indicates how important it is to fulfill its current needs. Urgency is defined as a function $u_m: S \rightarrow \mathbb{R}$, which expresses how urgent a particular motivator m's needs are in the state s_c.

The Goal Formulator accesses the Planner outside the GDA cycle to compute a plan and expectations for every available goal g. While this can prove expensive in some domains, ARTUE's use of an HTN planner allows us to exploit the relatively low cost of hierarchical planning (Ghallab *et al*. 2004), and feasible goals do not require more than a few seconds of planning per goal in our tests.

Each motivator then evaluates the *fitness* of each goal g for satisfying its domain-independent needs by applying a motivator-specific fitness function $f_m: \langle x_c, x_{c+1}, ... x_{c+n} \rangle \rightarrow \mathbb{R}$ to the expectations $x_{c+1}, ... x_{c+n}$ generated by the Planner. Finally, for each goal, a weighted sum over the motivators is calculated, defined as:

$$fitness(g) = \sum_m u_m(s_c) * f_m\big(Expectations(g, s_c)\big),$$

where g is a goal and *Expectations(g,s_c)* is the list of expectations X returned by the Planner when given a goal g in state s_c. The goal g with the highest *fitness(g)* is selected.

In the following sections, we describe the characteristic urgency and fitness functions for each of the three motivators. Specific experimental values for constants are discussed in Section 5.

4.1 Social Motivator

Urgency
The Social Motivator attempts to satisfy the desires of external entities. Currently, these are represented by a list of state conditions (which must be true to satisfy a desire) provided to M-ARTUE. Treating externally-provided goals as comparable to internally-formulated goals reflects our belief that an agent should be able to override or assign lower priority to goals given to it by a human. This may be useful because it allows humans to supply goals without considering an agent's internal needs.

The Social Motivator's urgency is a sawtooth function that increases over time until social desires are fulfilled. This function biases goal formulation toward social conditions when they have not been achieved in some time. It is defined by the function:

$$u_{social}(s_c) = \begin{cases} C_{social} u_{social}(s_{c-1}), & if\ R(s_c) \leq R(s_{c-1}) \\ 0.1, & if\ R(s_c) > R(s_{c-1}) \end{cases},$$

where s_c is the state at time of goal formulation, $R(s_c)$ is the percentage of user-provided goals that have been satisfied in s_c or some prior state s_i ($i<c$) visited by M-ARTUE, and $C_{social} > 1$ is a constant of social motivation that is tuned to the domain. This function biases the motivator to continually increase the number of desired state conditions that are satisfied, but the desire to do so is decreased directly after social progress is made.

Fitness
Intuitively, the value of a goal to the Social Motivator is measured by how many of the externally-provided desired state conditions are achieved when pursuing that goal. Therefore, the fitness function for the Social Motivator biases goal formulation toward goals that achieve the most social conditions with the fewest actions. It is calculated as:

$$f_{social}(X) = C_{social-fitness} \frac{R(x_{c+n}) - R(s_c)}{n},$$

where X is the sequence of expected states as defined above, x_{c+n} is the expected state after the plan executes, and n is the plan's length.

4.2 Exploration Motivator

Urgency
The Exploration Motivator chooses goals expected to best expand the agent's world knowledge by visiting unexplored states. This differs from the typical tradeoff between exploration and exploitation found in RL agents (Sutton and Barto 1998), as it permits the agent to plan paths to unseen states, and biases the agent toward unexplored states rather than random actions. For instance, a robot may be able to see more of its surrounding territory, locating dangers and new resources, by exploring.

The urgency of the Exploration Motivator is biased to increase when the latest action has not visited a new unique state, and to be large when fewer states overall have been visited (i.e., exploration is most valued when little exploration has been done). It is defined as:

$$u_{exploration}(s_c) = 1 - \frac{V(s_0, s_1, \dots, s_c)}{V(s_0, s_1, \dots s_{c-1}) + C_{exploration}},$$

where $V(S)$ is the number of distinct states in a list S and $C_{exploration}$ is a constant of exploration that is tuned to the domain.

Fitness
The Exploration Motivator values a goal in proportion to the number of previously unvisited states that M-ARTUE visits during plan execution. Therefore, its fitness function biases goal selection toward goals that visit the most new unique states per action. This function is defined as:

$$f_{exploration}(X) = \frac{V(s_0, s_1, \dots s_c, x_{c+1}, x_{c+2}, \dots x_{c+n}) - V(s_0, s_1, \dots s_c)}{n}.$$

4.3 Opportunity Motivator

Urgency
The Opportunity Motivator tries to maximize the agent's opportunity to act throughout plan execution, thus helping the agent to prepare to fulfill future goals. We evaluate this need in terms of two factors. The first is the expected number of actions available to M-ARTUE from a given state, which represents its ability to react quickly in unexpected situations and fulfill new goals. The second is the availability of resources relative to their historical averages, where a resource is a finite, real-valued quantity that is reduced to perform actions as specified in the domain (e.g., fuel used in navigation). These factors are combined to determine this motivator's urgency, which biases formulation toward opportunity-increasing goals when the agent cannot execute as many actions or does not possess as many resources as have been available historically. This function is defined as:

$$u_{opportunity}(s_c) = \left[\left(1 - \frac{N(s_c)}{max_{0 \leq i < c} N(s_i)}\right) + (1 - L(s_c))\right]/2,$$

where $N(s)$ is the number of available actions, and $L(s)$ is the level of resources relative to historical resource levels. A domain defines a set of k resources, each of which has a state-based level $v_r(s)$. Function $L(s)$ is defined in terms of these levels as $L(s_c) = (\sum_{r=1}^{k} [v_r(s_c)/a_r(s_c)])/k$, where $a_r(s_c) = \frac{\sum_{i=1}^{c-1} v_r(s_i)}{c-1}$ is the mean of all prior values for $v_r(s)$.

Fitness
When evaluating the agent's opportunities, the motivator should recognize the long-term repercussions of satisfying

a plan's goal. For instance, after the goal of recharging a robot is fulfilled, it can act for a longer time. Therefore, the Opportunity Motivator's fitness biases goal formulation toward goals that have the most actions available per expected state, and leave the agent with the most resources and actions available when the goal is achieved. This function is defined as:

$$f_{opportunity}(X) = \frac{\left(\left[\sum_{j=0}^{n-1} N(x_{c+j}) \right] + [w * N(x_{c+n})] \right)}{(n+w)N(s_c)} + L(x_{c+n}) - L(s_c) - 1$$

where $w \geq 1$.

5. Empirical Study

5.1 Rover Domain

Rovers-With-Compass (RWC) is a deterministic navigation domain with hidden obstacles inspired by the difficulties encountered by the Mars Rovers. In this domain, individual locations may be windy, sandy, and/or contain sand pits, but the agent cannot observe these obstacles directly. Sandy locations cause the rover to be covered in sand. While covered in sand, the rover cannot observe its location or recharge its batteries. Sand pits stop the rover from moving, but the rover can dig itself out at a high energy cost. Windy locations clear the sand off of the rover, but due to a malfunction, may confuse the rover's compass, causing it to move in the wrong direction. Rovers in this domain have a finite amount of energy that is consumed by actions; if a rover runs out of energy, it becomes unable to move. Success in this domain is evaluated based on the agent's ability to achieve a set of three separate navigation goals using different rovers.

5.2 Experimental Design

Our primary claim is that M-ARTUE should perform comparably to ARTUE, as measured by the percentage of goals successfully achieved, despite its smaller amount of task-specific knowledge. As a secondary claim, we intend to show that both the Opportunity and Exploration Motivators improve the performance of M-ARTUE. In order to measure goal achievement performance, we tested ARTUE and M-ARTUE using three sets of 25 randomly-generated scenarios within the RWC domain. Each scenario consisted of a grid of 36 locations; each location had a fixed probability of being windy, being sandy, being sunny, and containing a sand pit. These probabilities varied across three experimental conditions and correspond to the "danger" of the domain. M-ARTUE was responsible for controlling three rovers in each scenario, and each rover had its own goal location, but the agent could exploit knowledge gained using one rover in planning for another. M-ARTUE's objective (and social motivation) was to get each rover to its goal location in the scenario, and performance was measured as a percentage of goal locations reached. Each scenario was tested with both M-

ARTUE and ARTUE, which used the same HTN definitions for goal planning. A limit of 80 actions was imposed on both agents to ensure the timely conclusion of all tests. (Allowing more actions could only improve M-ARTUE's relative performance, as ARTUE completed every scenario in fewer than 80 actions.)

M-ARTUE chose goals using the heuristic-guided goal formulation process described in Section 4. Available goals considered by M-ARTUE included recharging, navigation, and recovery (i.e., removing a rover from a sand-pit); successful completion of scenarios required that all types of goals be used in different situations. ARTUE's formulation rules asserted recharge goals for partially-discharged rovers, and navigation goals for rovers not at their targets. The priority of the recharge goal varied based on the amount of remaining charge. ARTUE always selected the highest priority goal for which it could find a plan. Both agents planned with SHOP2, using a mapping from goals into tasks and standard hierarchical task decomposition using manually designed hierarchical task methods. In this experiment, C_{social} was set to 1.1, giving a modest growth rate for social urgency. $C_{social-fitness}$ was set to 5, slightly larger than the average plan length, to keep social fitness largely within the range of [0,1]. $C_{exploration}$ was set to 5, to make exploration most important during the first 5 actions. Finally, w was set to 20, to ensure final states outweighed the rest of longer-than-average plans.

5.3 Results

In our first test, the probability of obstacles was set to 10%, so there was a 10% probability of windy conditions, sandy conditions, and a pit at each location. With 50% likelihood, each location is sunny, meaning the rover can recharge there. Our second test used probabilities of 20% and 40%, respectively, and our third test used 30% and 30%. As a result, successive tests were more dangerous.

Figure 2 compares the performance of M-ARTUE to ARTUE for each of our three test conditions. In the first test, M-ARTUE actually outperformed ARTUE, although the difference is not significant (p=.07). For the second and third tests, M-ARTUE and ARTUE did not perform significantly differently, supporting our claim that domain-independent heuristics can perform at the same level as engineered rules for goal formulation.

In Figure 3, we compare the performance of two ablations in our three test conditions. As expected, performance suffers without the Opportunity Motivator. In each test condition, the full M-ARTUE significantly outperforms M-ARTUE without the Opportunity Motivator (p<.05). The same is not true for the Exploration Motivator. While there is a significant benefit in the first test (p=.03), this decreases as obstacles become more frequent, which matches our intuition that exploration is less useful in more dangerous conditions, supporting our claim that both motivators contribute to superior performance, at least some of the time.
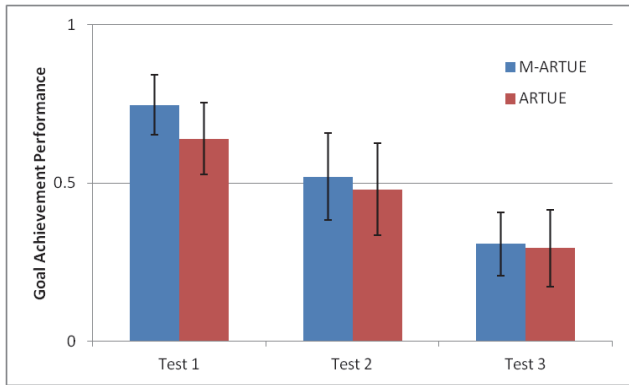
Figure 2: Comparing M-ARTUE vs. ARTUE

Figure 3: Comparing M-ARTUE vs. ablations

6. Conclusions and Future Work

While our experiments support our claim that domain-independent heuristics can, for some tasks and domains, replace hand-coded knowledge in goal formulation, much work remains to be done. First, it's important to identify the generality of these techniques in applications to other domains. Second, our experiments required tuning constants in the motivator functions (see Section 4.1). This use of domain-specific knowledge is undesirable and we plan to instead automatically tune them in our future work. Third, we intend to address the higher overhead of planning for all goals on every GDA cycle, possibly by filtering goals that can be identified *a priori* as non-contributory. Fourth, resources are presently identified as part of the domain description, but could be algorithmically discovered in future work. Finally, we plan to replace the Social Motivator with an interactive system that allows M-ARTUE to learn goal formulation knowledge, as discussed in prior work (Powell *et al.* 2011), to permit long-lived agents to adapt their formulation policies over time.

Acknowledgements

References

Bratman, M.E. (1987). *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Choi, D. (2011). Reactive goal management in a cognitive architecture. *Cognitive Systems Research*, **12**, 293-308.

Coddington, A.M. (2006). Motivations for MADbot: a motivated and goal directed robot. *Proceedings of the Twenty-Fifth Workshop of the UK Planning and Scheduling Special Interest Group* (pp. 39-46). Nottingham, UK: University of Nottingham.

Cox, M.T., & Veloso, M.M. (1998). Goal transformations in continuous planning. In M. desJardins (Ed.), *Proceedings of the Fall Symposium on Distributed Continual Planning* (pp. 23-30). Menlo Park, CA: AAAI Press.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory and Practice*. San Francisco, CA: Morgan Kaufmann Publishers.
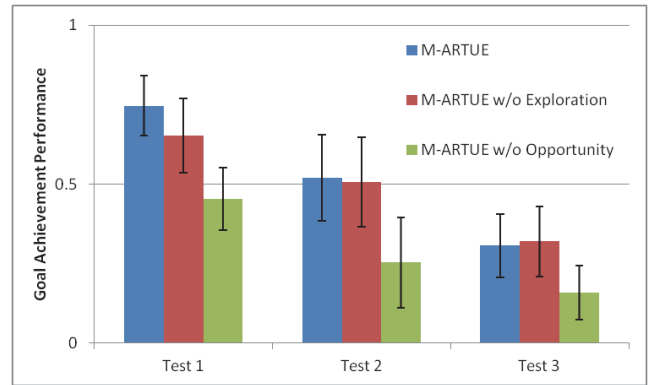
Hawes, N. (2011). A survey of motivation frameworks for intelligent systems. *Artificial Intelligence*, **175**(5-6), 1020-1036.

Hawes, N., Hanheide, M., Hargreaves, J., Page, B., Zender, H., & Jensfelt, P. (2011). Home alone: Autonomous extension and correction of spatial representations. In *ICRA* (pp. 3907-3914). Shanghai, China: IEEE Press.

Jaidee, U., Munoz-Avila, H., & Aha, D.W. (2011). Integrated learning for goal-driven autonomy. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. Barcelona, Spain: AAAI Press.

Klenk, M., Molineaux, M., & Aha, D.W. (2012). Goal-driven autonomy for responding to unexpected events in strategy simulations. To appear in *Computational Intelligence*.

Molineaux, M., Klenk, M., & Aha, D.W. (2010a). Goal-driven autonomy in a Navy strategy simulation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Molineaux, M., Klenk, M., & Aha, D.W. (2010b). Planning in Dynamic Environments: Extending HTNs with nonlinear continuous effects. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Molineaux, M., Kuter, U., & Klenk, M. (2012). DiscoverHistory: understanding the past in planning and execution. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multi-Agent Systems*. Valencia, Spain: ACM Press.

Nau, D.S. (2007). Current trends in automated planning. *AI Magazine*, **28**(4), 43–58.

Nau, D., Au, T.-C., Ilghami, O, Kuter, U, Murdock, J.W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, **20**, 379-404.

Powell, J., Molineaux, M., & Aha, D.W. (2011). Active and interactive learning of goal selection knowledge. In *Proceedings of the Twenty-Fourth Florida Artificial Intelligence Research Society Conference*. West Palm Beach, FL: AAAI Press.

Sun, R. (2007). The motivational and metacognitive control in CLARION. In W. Gray (Ed.) Modeling integrated cognitive systems (pp. 63–75). New York: Oxford University Press.

Sutton, R.S., & Barto, A.G. (1998). *Introduction to Reinforcement Learning*. Cambridge, MA: MIT Press.

Weber, B., Mateas, M., & Jhala, A. (2012). Learning from demonstration for goal-driven autonomy. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence.* Toronto, Canada: AAAI Press.